Call vs Apply vs Bind

What is "this"?

తెలుగు లో JavaScript

Venkatesh Mogili
#WebGuru

What is "this" in JavaScript?

- **"this"** refers to an object where it called.
- It depends on it's context completely.

- Example:
- If we call this inside an object, it's context will be that particular object.

```
let person={
    firstName:"Venkatesh",
    getName:function(){
        console.log(this.firstName)
    }
}
person.getName(); // Venkatesh
```

Real time example

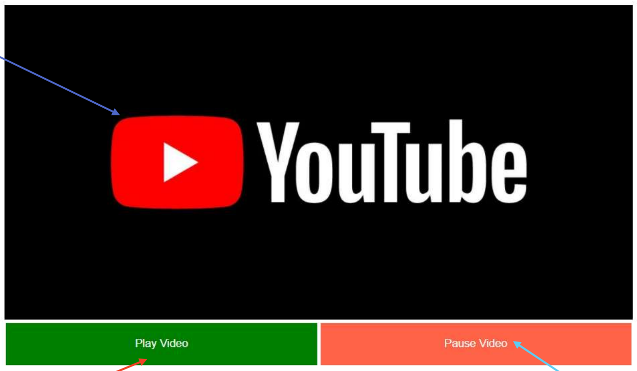
“this” refers to ball

“this” refers to house

“this” refers to kid

Real time application example

“this” refers to video



“this” refers to play button

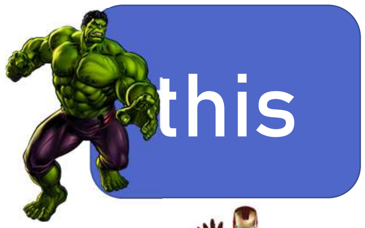
“this” refers to pause button

Basic rules to remember

- this always **refers to an object**
- this refers window object in **global context**
- this refers to an object *where we call it*.

We are losing the actual context of "this" based on the place where we are calling. So, how can we rescue from this?

Simple example to rescue "this"



Bind

Call

Apply

Bind

1. **Bind** method creates a new function and sets the “this” keyword to the specified object.
2. We can pass second argument as optional arguments with comma separated values.

3. Syntax:
   function.bind(**thisArg**,optionalArguments)

4. Realtime example in React components:

```
Class ComponentName extends Component{
    this.submit=this.submit.bind(this);
    this.handleChange=this.handleChange.bind(this);
    submit(){}
    handleChange(){}
}
```

Call

1. **Call** method won’t create new function, instead it will executes immediately by setting the “this” keyword to the specified object.
2. We can pass second argument as optional arguments with comma separated values.

3. Syntax:
   function.call(**thisArg**,optionalArguments)

4. Example:

```
let data={name:”Iron Man”}
function greeting({name}){console.log(name);}
greeting.call(data);
```

Apply

1. **Apply** method also won't create new function similar to call method and it will executes the function immediately by setting the "this" keyword to the specified object.
2. We can pass second argument as **Array of optional arguments** instead of comma separated values, remaining everything is similar to call method.

3. Syntax:
   function.apply(**thisArg**,optionalArguments)

4. Example:
   let strongestAvenger=[1,2,3];//1.Spider man, 2.Iron man, 3.Avengers
   console.log(Math.max(strongestAvenger))? ❌
   console.log(Math.max.**apply**(null,**strongestAvenger**)) ✓

Summary

1. Bind **creates a new function** and sets "this" keyword to the specified object.

2. Call and apply methods won't create a new function, instead they will **execute immediately** and sets "this" keyword to the specified object.

3. Apply method will take only **array of arguments** as the second parameter instead of comma separated values.

Don't forget to Share and Subscribe

LIKE SHARE COMMENT

SUBSCRIBE

Thank you for watching

Subscribe

and

Stay Tuned

For more updates…

VENKATESH MOGILI

#Web Guru